# ModCoupler-Verilog

## User's Guide

**Powersim Inc.**

# ModCoupler-Verilog User's Guide

**Version 1.0**

Release 1.0

October 2012



## Disclaimer

Powersim Inc. ("Powersim") and the Carlos III University of Madrid make no representation or warranty with respect to the adequacy or accuracy of this documentation or the software which it describes. In no event will Powersim and the Carlos III University of Madrid or its direct or indirect suppliers be liable for any damages whatsoever including, but not limited to, direct, indirect, incidental, or consequential damages of any character including, without limitation, loss of business profits, data, business information, or any and all other commercial damages or losses, or for any damages in excess of the list price for the license to the software and documentation.

**Powersim Inc.**

email: info@powersimtech.com

http://www.powersimtech.com

# 1 Introduction

ModCoupler-Verilog is a communication link between the simulation software ModelSim® and PSIM®. By means of ModCoupler-Verilog, the co-simulation of the completed power electronics equipment can be performed. The digital control algorithm, described in Verilog, will be simulated in ModelSim and the power stage will be simulated in PSIM.

On the PSIM side, one must include the ModCoupler-Verilog block in the schematic. On the ModelSim side, no changes are needed. From PSIM´s schematic, the user will provide information to ModCoupler-Verilog module on time-step, Verilog clk signal frequency and input-output signal information to configure the co-simulation. Fig. 1 shows the basic co-simulation structure.

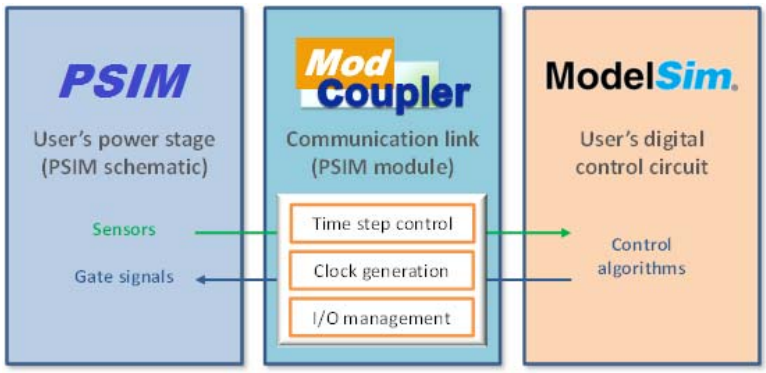


**Fig. 1: Co-simulation architecture.**

At every simulation instant both simulators are stopped. The input signal values of the ModCoupler module are forwarded to the digital circuit, which is in charge of the control algorithms. Once ModelSim´s calculation is finished, the Verilog output values are sent back to close the loop. After that, each simulator runs a new simulation step and the cycle is repeated. Note that one PSIM simulation step correspond to several ModelSim simulation steps.

This guide describes how to build a co-simulation environment from scratch. It is assumed that the user provides a PSIM schematic for the analog simulation and a Verilog description for the digital simulation.

## 2 ModCoupler-Verilog block configuration

In your PSIM schematic file (create a new one if it is needed) add a ModCoupler-Verilog block (placed on the *Control* submenu of the *Elements* menu) as show in Fig. 2.

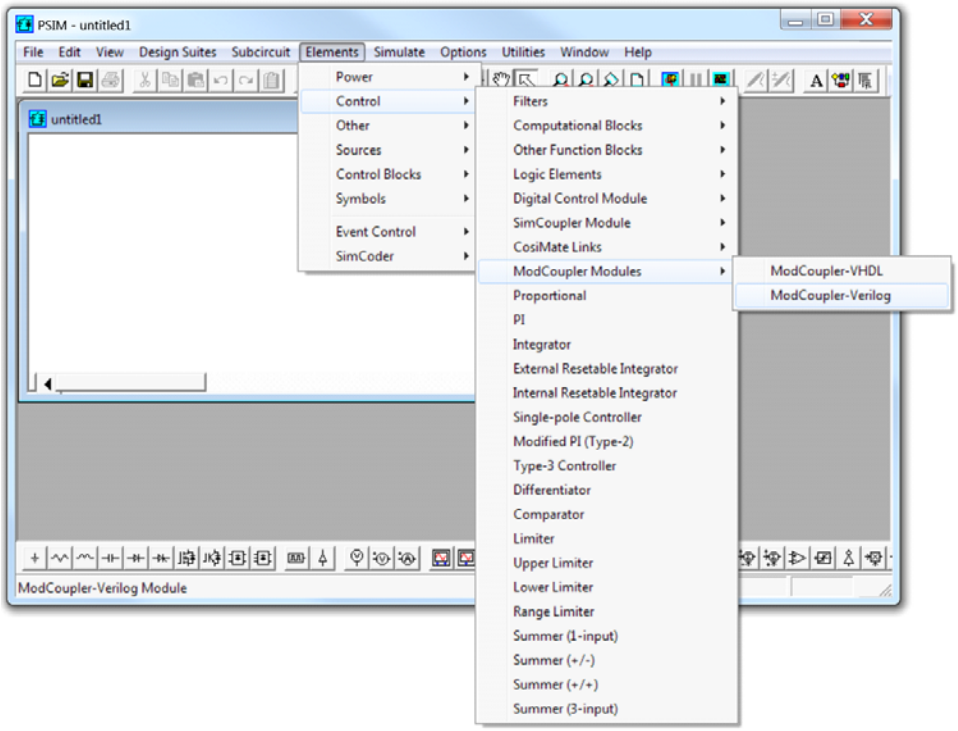


**Fig. 2: ModCoupler-Verilog module menu**.

In Fig. 3 the main dialogue window of the ModCoupler-Verilog module is shown.

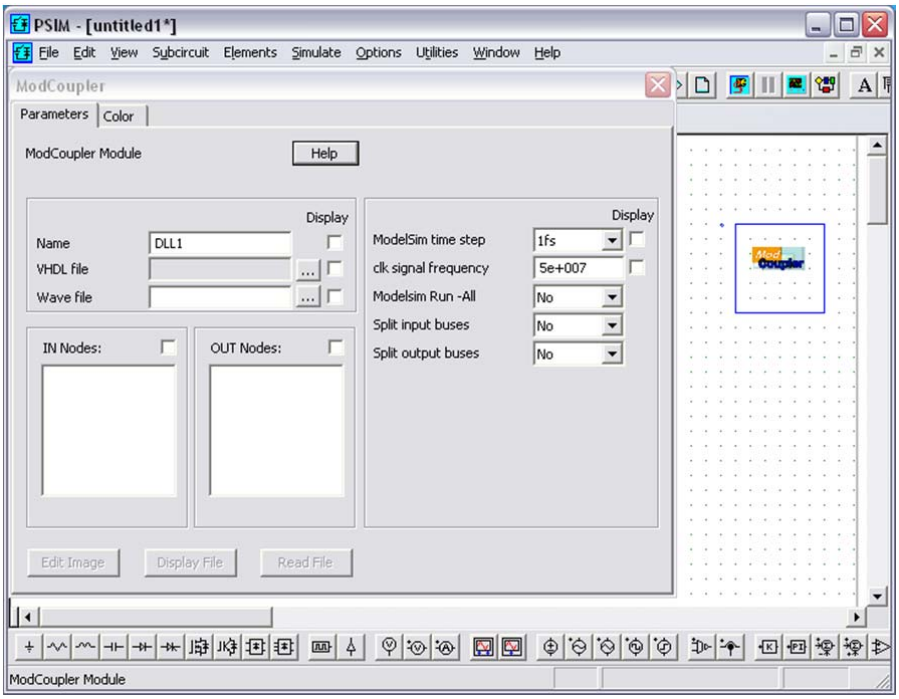


**Fig. 3: ModCoupler-Verilog module dialog window.**

The different parameters are explained below:

- *Verilog file:*The .vhd top entity file. After this file is selected, the IN/OUT nodes lists will be created.
- *Wave file:* The file with the signals to be displayed in ModelSim waveform window.
- *ModelSim Time Step:* The ModelSim simulation time step. This value must be smaller than the PSIM time step of the Simulation Control
- *Clk signal frequency:* The frequency of the ModelSim clk signal. Notice that the clk signal is not processed as an input, since the period is usually smaller than the PSIM time step.
- *ModelSim Run All (Yes/No):* Allows starting the ModelSim simulation without pressing the "Run –All" button. It is recommended to set that parameter to "No" the first time the simulation is run in order to select the signals to be displayed in ModelSim waveform window.
- *Split input buses (Yes/No):* Allows splitting an input vector signal in its different bits.
- *Split output buses (Yes/No):* Allows splitting an output vector signal in its different bits.

# 3 Compilation of Verilog files

ModCoupler-Verilog needs a compiled model of the Verilog design in order to operate. Also, the "work" ModelSim library must be located in the working directory (it can be done easily moving the *work* folder to the directory that contains the schematic file).

The proposed compilation method uses a Windows batch file, although this process should be performed by the ModCoupler-Verilog dialog. A batch file includes commands to be executed. Example batch files (compile.bat) can be found in the examples directories. ModelSim applications *vcom* and *vlib* are used, so the path to both of them must be in the environment variable PATH.

NOTE: If any change is made on any Verilog file, the model must be recompiled.

# 4 Simulation

The last step is running the simulation. Set the configuration ModelSim Run All parameter to "No" the first time the simulation is run, so ModelSim opens without actually running the simulation. In this state, the user can select the appropriate signals to display and save them to the wave file (wave.do by defect).

Start the simulation pressing the PSIM "Run simulation engine" button. After a few seconds, a ModelSim window will appear. Once the user has selected the signals to view, the simulation is started by pressing the "Run –All" button in ModelSim.

In next simulations, the *ModelSim Run All* parameter can be set to "Yes" to start ModelSim and run the simulation automatically.

If a new simulation with the same Verilog model is needed (for example, after editing a Verilog file and recompiling or after a schematic change), press the ModelSim "Restart" button before pressing the PSIM "Run simulation engine" button again (closing ModelSim window is not required).

# 5 Example: Current loop of a buck converter

## 5.1 ModCoupler block configuration

Before configuring the ModCoupler block, the PSIM schematic will be created.

In this example PSIM is in charge of the buck power simulation while ModelSim® is doing the control part. The final schematic is included in the working directory (*examples\Verilog*) and its name is buck_iL.psimsch but, in this example, the complete communication process between PSIM and ModelSim is explained. So, we will start with a buck converter (Fig. 4).

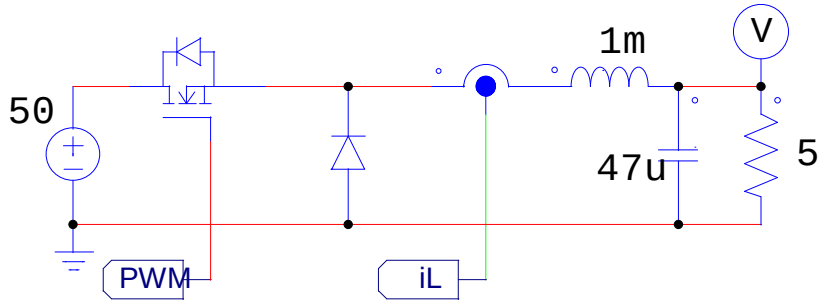


**Fig. 4: PSIM´s schematic of a buck converter.**

To make this design functional it's only necessary to give PSIM the MOSFET pulses. These pulses will be calculated by means of the ModelSim simulator. For this purpose we'll add a ModCoupler block and select Buck_iL_Loop.v (located in the *verilog* folder of this example) as HDL file (Fig. 5).The IN/OUT Nodes lists will be created.

The output will be obviously connected to the MOSFET gate in order to commutate it. The 3 different inputs are the following:

- Reset: The reset signal of the digital design.
- iL: The inductor current.
- iLref: The current reference.

Next, the ModCoupler Block parameters will be set with the following values:

- *Wave file*: Point to the wave.do file placed on the example folder.
- ModelSim time step: 10ns
- Clk signal frequency: 1e+007
- ModelSim Run All: Yes
- Split input buses: No
- Split output buses: No

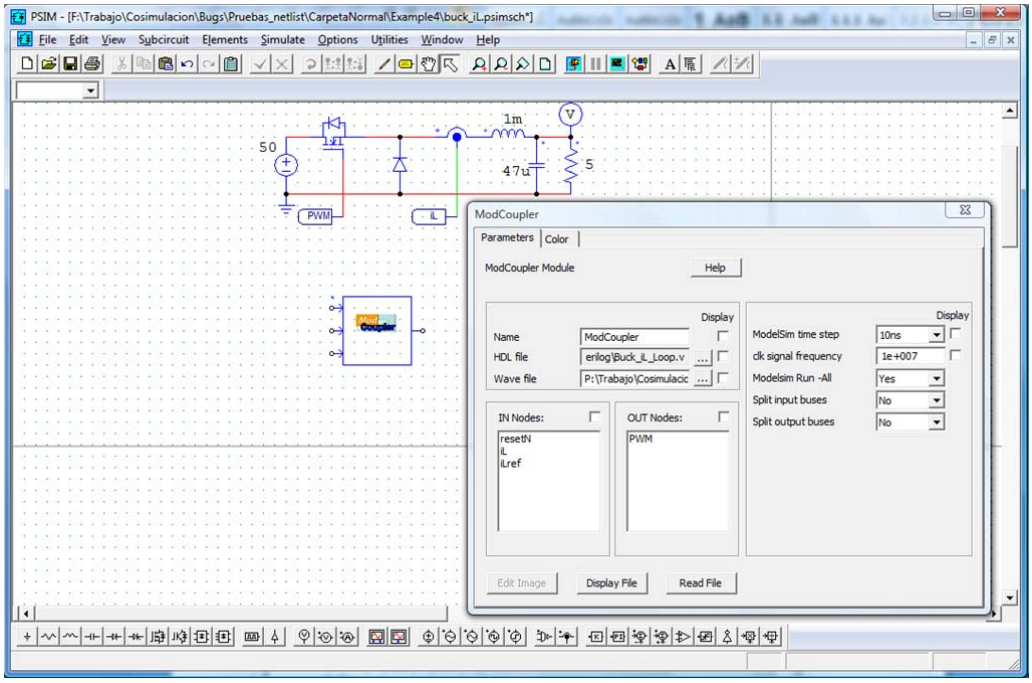


**Fig. 5: ModCoupler Block dialog window.**

This Verilog model only uses integer variables in order to make it closer to a real model, so a quantization block is required to make the real(PSIM) to integer(ModelSim) transformation. This quantization block is equivalent to an ADC, but it gives all the bits together instead of the bits separately.

The final schematic can be seen in Fig. 6

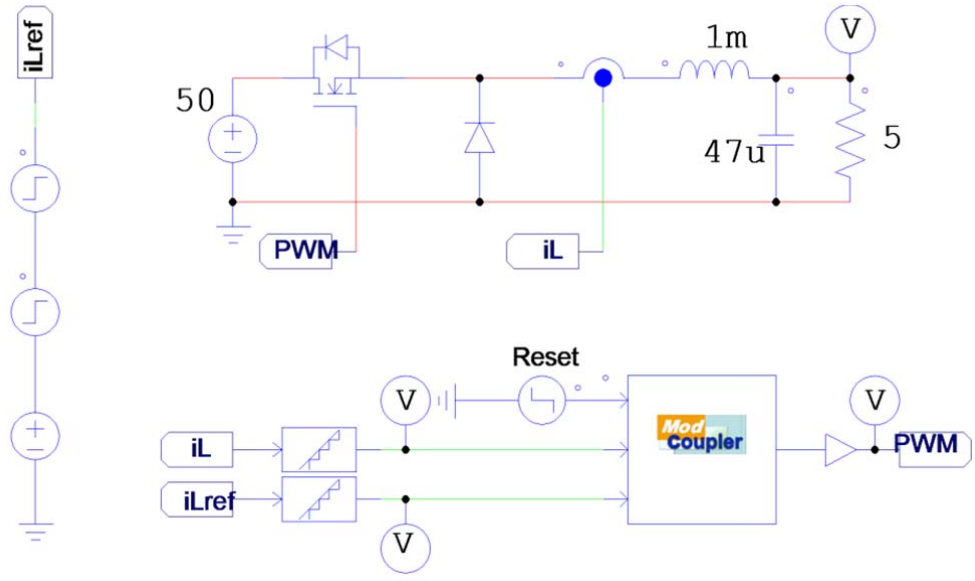


**Fig. 6: Schematic of the buck converter example.**

## 5.2 Compilation of Verilog files

Since there is only one Verilog file in this example, it is not needed having the model pre-compiled. Anyway, doing the compilation is recommended to see the possible errors of the

Verilog design. For this purpose a batch file (compile.bat) has been included with an example of compilation.

## 5.3 Simulation

The last step is run the simulation. Start it by pressing the PSIM "Run simulation engine" button. At this point, ModCoupler creates a Verilog file called ModCouplerTemporaryFile.v in the Verilog directory and compiles it. After a few seconds, a ModelSim window will appear with the compiled model. As the RunAll parameter was set to "Yes", the ModelSim simulation will start immediately.

*NOTE*: ModCoupler uses the ModelSim applications *vcom* and *vsim*, so the path to both of them must be in the environment variable PATH

## 5.4 Restarting the simulation

If there are no changes in the Verilog model, a restart can be easily done by following these steps:

1. Restart the ModelSim model clicking in the restart button (Fig. 7).
2. Push the Run Simulation button on PSIM.
3. Push the "Run –all" button on ModelSim (the "run –all" option of ModCoupler only works if ModelSim was closed in the second step)


**Fig. 7: ModelSims's Simulate toolbar.**

## 5.5 Example: Comparison between analog and Verilog control.

In this example a comparison between the previous example and the analog equivalent is done. The schematic is located in the folder *examples\Verilog_vs_Analog* and is shown in Fig. 8.

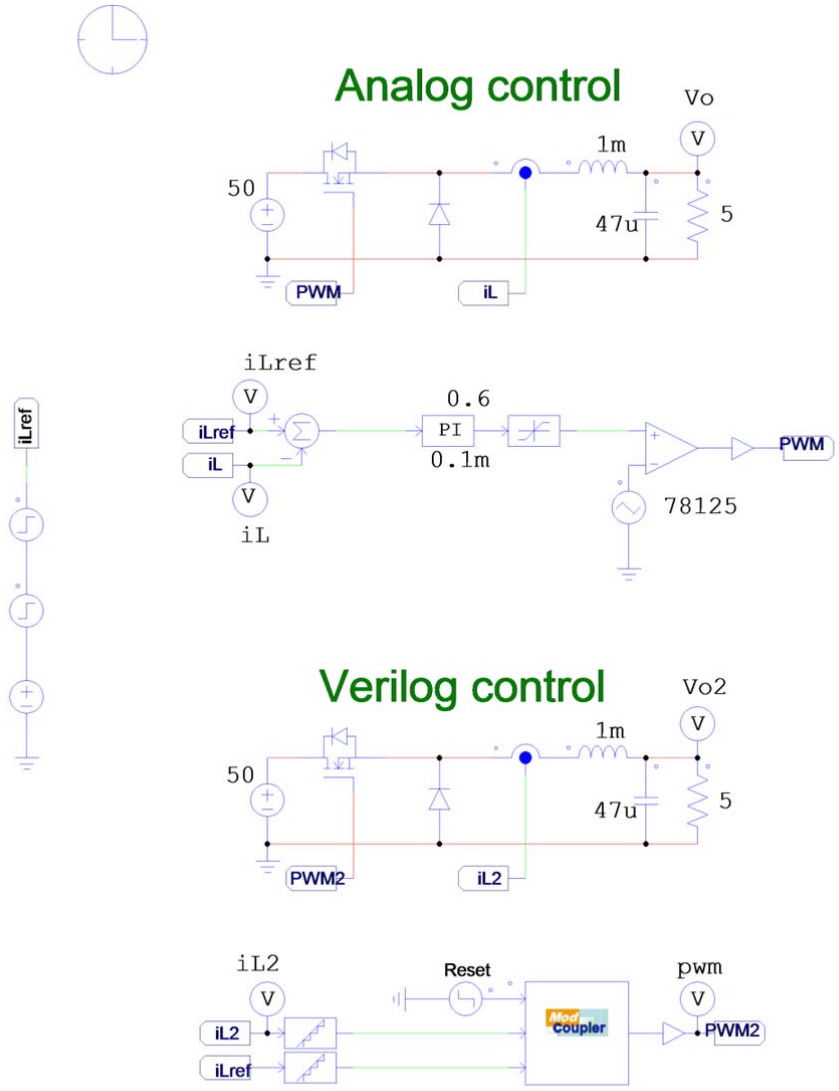


**Fig. 8: Schematic of the comparison between an analog and a Verilog control.**

The Verilog control was done from the analog one, performing the discrete model of the PI regulator. The results of the comparison are shown in Fig. 9, where the red traces represent the analog values, the blue traces represent the Verilog values and the green trace is the current reference.

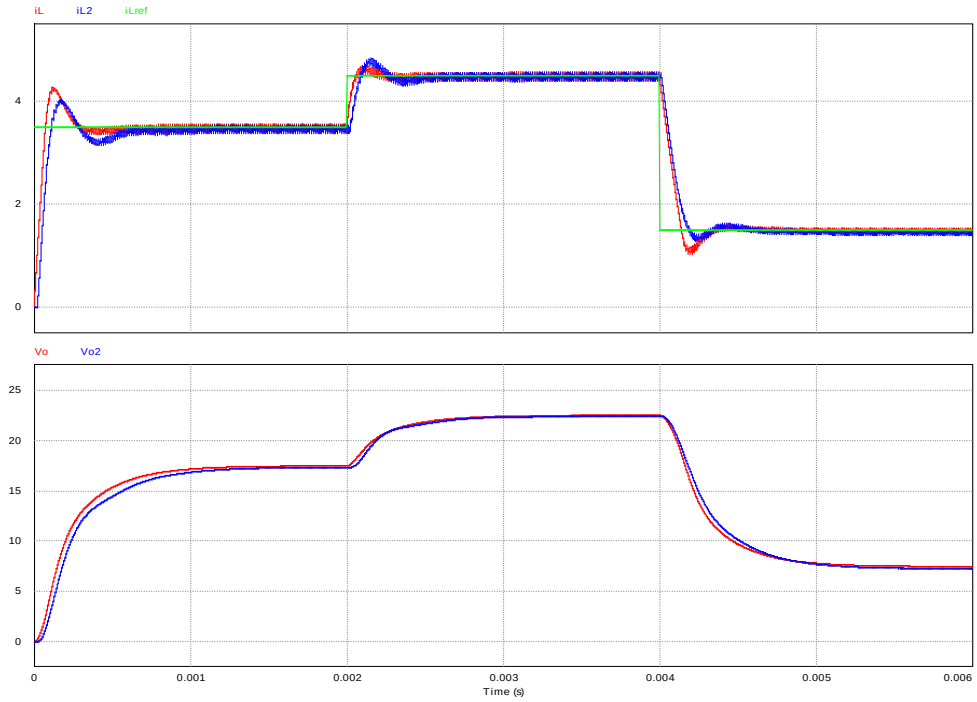


**Fig. 9: Simulation comparison result**

# 6 Error messages

ERROR 001: “vlib cannot be executed”: This error may occur if the vlib binary cannot be executed. Try using ModCoupler from an administrator Windows account.

ERROR 002: “vlog cannot be executed”: This error may occur if the vlog binary cannot be executed. Try using ModCoupler from an administrator Windows account.

ERROR 003: “vsim cannot be executed”: This error may occur if the vsim binary cannot be executed. Try using ModCoupler from an administrator Windows account.

ERROR 004: “The maximum creating process time has been exceeded": ModCoupler waits one minute to the ModelSim process to be created. If this time is exceeded the simulation is cancelled. Try freeing memory for a better performance of the computer.

ERROR 005: “The pipe cannot be opened”: The communication pipe cannot be opened. Try restarting the computer.

ERROR 006: “The header cannot be sent”. Report the error to Powersim.

ERROR 007: “The name size cannot be sent”. Report the error to Powersim.

ERROR 008: “The name cannot be sent". Report the error to Powersim.

ERROR 009: “The name size cannot be sent". Report the error to Powersim.

ERROR 010: “The name cannot be sent". Report the error to Powersim.

ERROR 011: “The HDL directory cannot be sent". Report the error to Powersim.

ERROR 012: “The .do file cannot be sent". Report the error to Powersim.

ERROR 013: “The maximum creating pipe time has been exceeded". ModCoupler waits one minute to the communication pipe to be created. If this time is exceeded the simulation is cancelled. Try freeing memory for a better performance of the computer.

ERROR 014: “ModelSim binary files not found. Reinstall ModelSim or include it in the PATH system environment variable".

ERROR 015: “ModelSim time step is higher than clk signal period, clk signal cannot be generated". Decrease the ModelSim time step.

ERROR 016: “The ratio between the PSIM time step and the ModelSim time step is less than one. Avoid using folders with spaces in the name. It's recommended decreasing ModelSim time step": This error occurs when using a folder with spaces in the name or when the ratio between PSIM and ModelSim time steps is less than one.

ERROR 017: “The pipe cannot be found": The communication pipe cannot be found. Try restarting the computer.

ERROR 018: “The pipe cannot be opened": The communication pipe cannot be opened. Try restarting the computer.

ERROR 019: “Unable to write to the pipe": The communication pipe failed. Try restarting the computer.

ERROR 020: “Unable to read from the pipe": The communication pipe failed. Try restarting the computer.